TITLE:

Sparse Bayesian learning of filters for efficient image expansion

AUTHOR(S):

Kanemura, Atsunori; Maeda, Shin-ichi; Ishii, Shin

CITATION:

Kanemura, Atsunori ...[et al]. Sparse Bayesian learning of filters for efficient image expansion. IEEE transactions on image processing 2010, 19(6): 1480-1490

ISSUE DATE:

2010-06

URL:

http://hdl.handle.net/2433/123421

RIGHT:

(c) 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

京都大学学術情報リポジトリ
KURENAI
Kyoto University Research Information Repository

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

Sparse Bayesian Learning of Filters for Efficient Image Expansion

Atsunori Kanemura, *Member, IEEE*, Shin-ichi Maeda, and Shin Ishii

***Abstract*—We propose a framework for expanding a given image using an interpolator that is trained *in advance* with training data, based on sparse Bayesian estimation for determining the optimal and compact support for efficient image expansion. Experiments on test data show that learned interpolators are compact yet superior to classical ones.**

***Index Terms*—Automatic relevance determination (ARD), image expansion, image interpolation, resolution synthesis (RS), sparse Bayesian estimation, variational estimation.**

I. Introduction

CLASSICAL methods for image expansion such as bilinear interpolation or splines can be understood as linear filtering operations on a given image, and their support and coefficients are designed based on top-down assumptions, e.g., the image is a piecewise polynomial and smooth at the knots. However, these assumptions are not necessarily true for natural images. Alternatively, the support and coefficients of the filter can be *learned from real image data*. Arguably, learning-based approaches can yield better performance than top-down strategies [1]–[3]. In principle, a learning-based filter design can use arbitrary size support. This is in contrast to the bilinear interpolator, which uses at most four low-resolution pixels when determining the value of a pixel in the high-resolution expanded image. The support should be simple for efficient processing of the images and for preventing overfitting; however, excessively simple ones will fail to capture the useful information contained in the surrounding pixels. The compactness of the support is beneficial when we want a fast and high-quality image interpolator, especially when we apply it in small embedded systems such as digital cameras and mobile phones. In this paper, we aim to resolve the tradeoff between high quality and low cost.

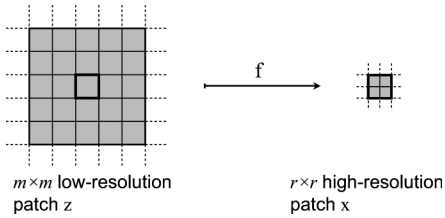


Fig. 1. $m \times m = Q$ low-resolution pixels are used to estimate $r \times r = D$ high-resolution pixels. This figure depicts a case where $m = 5$ and $r = 2$.

Let r be an integer magnification factor. The task of image expansion is:

given an $M \times N$ image $\boldsymbol{\xi}$, estimate $rM \times rN$ expanded image $\hat{\boldsymbol{\xi}}$.

In our framework, the interpolator expands the image by replacing each pixel in the given low-resolution image by an $r \times r$ high-resolution image patch. Of course, since estimating r^2 pixel values is impossible from only one pixel value, we use the low-resolution pixel *patch* surrounding the pixel to be replaced (Fig. 1). This local interpolation is repeated for every pixel in the given image, and the expanded image is constructed by tessellating the high-resolution patches. Vector-valued function $\mathbf{f}$ maps an $m \times m$ low-resolution patch to an $r \times r$ high-resolution patch.

We address the problem of determining optimal supports by formulating the image interpolation task from a viewpoint of sparse Bayesian estimation. A simple method to determine the optimal shape of the support would be to perform discrete optimization that compares 2^{m^2} different shapes of the support. Obviously, this approach soon becomes intractable when m gets larger. Alternatively, sparse Bayesian methods [4]–[7] offer continuous parameters that regulate the importance of each pixel, and the less important pixels for the estimation of high-resolution patches are automatically pruned from the support of the filter.

The learning of filter coefficients has been considered by Triggs [8], emphasizing low-level vision and reducing aliasing, and by Atkins [1], whose proposal, called resolution synthesis (RS), uses a mixture of linear interpolators for image expansion. In [8], the interpolator is learned from pairs of the original images and their synthetically smoothed and subsampled images by optimizing several error metrics including L_1 and L_2 norms (which is equivalent to maximum-likelihood estimation). Triggs reported that the shapes of the learned interpolators resemble the sinc function and are robust to the change of error metrics or anti-aliasing smoothing kernels. He

Manuscript received July 22, 2008; revised January 14, 2010. First published March 08, 2010; current version published May 14, 2010. This work was supported by Grant-in-Aid for Scientific Research on Priority Areas, "Deepening and Expansion of Statistical Mechanical Informatics," from MEXT, Japan. The work of A. Kanemura was supported by Grant-in-Aid for JSPS Fellows 20-8856 and by Excellent Young Researchers Overseas Visit Program from JSPS. Preliminary results of this work were presented as an unreviewed presentation at the MIRU Workshop Subspace2008, Karuizawa, Japan, in July 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Arun Abraham Ross.

A. Kanemura is with the was with the Graduate School of Informatics, Kyoto University, Kyoto 611-0011, Japan, and the Department of Electrical Engineering, University of California, Santa Cruz, CA 95064 USA. He now with ATR Neural Information Analysis Laboratories, Kyoto 619-0288, Japan (e-mail: atsu-kan@sys.i.kyoto-u.ac.jp).

S. Maeda and S. Ishii are with the Graduate School of Informatics, Kyoto University, Kyoto 611-0011, Japan (e-mail: ichi@i.kyoto-u.ac.jp; ishii@i.kyoto-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2043010

1057-7149/$26.00 © 2010 IEEE

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

also investigated the influence of support size and found that the variation of test interpolation errors between $m = 3,\ 5,\ 7$ is significant, but beyond $m = 7$ the learned filters have similar test performance. Atkins's RS is modeled by a Gaussian mixture that is trained by maximum-likelihood estimation utilizing the expectation-maximization (EM) algorithm. Ni and Nguyen [9] refined RS by replacing linear interpolators with nonlinear support vector regressors. RS can be considered an image superresolution method (as in [9]) because its regressors contain information from external training data other than the given image. The support size of Atkins' original RS [1] is 5×5, but he did not provide logical justification for this choice. As far as we know, existing RS methods [1], [9]–[15] have not mentioned any reasonable way to determine the support size.

In comparison with superresolution methods, our framework, image expansion based on learned filters, can be understood as the simplest extreme of RS and is placed somewhere between RS, example-based superresolution methods that hallucinate a high-resolution image by searching patches in the example database [2], [3], and reconstruction-based superresolution methods that invert a generative model from a high-resolution image to multiple low-resolution images [16]–[18]. An important aspect of our framework is that the external information is encapsulated compactly in interpolator $\mathbf{f}$. Therefore, it does not suffer from the large computational loads required to use a mixture, to search through a large database, or to invert the forward optics; yet it is expected to have good performance based on the statistical integration of external data. Even though we do not argue that image expansion using learned filters is a superresolution method, it can be usable enough.

In Section II, we describe a linear regression model for predicting high-resolution patches from low-resolution patches and present maximum-likelihood and L_1-regularized estimation methods of image expansion filters. Section III introduces sparse Bayesian modeling, and in Section IV, we derive an iterative algorithm to efficiently solve the Bayesian estimation problem. Experimental results are presented in Section V, where we show that sparse Bayesian learning successfully obtains compact and efficient filters. Discussion on the modeling direction for estimating high-resolution images from low-resolution images is given in Section VI. Section VII summarizes this study.

II. Model and Basic Learning

Before the real image expansion jobs, we attempt to obtain $\mathbf{f}$ using a training dataset that consists of a large number of low- and high-resolution patches so that the filter learns the relationship between them. We regard these patches as lexicographically stacked vectors. Let $\mathbf{z}_n$ be the $m^2 = Q$-dimensional vectors of the low-resolution patches, let $\mathbf{x}_n$ be the $r^2 = D$-dimensional vectors of high-resolution patches, and let $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{z}_n)\}_{n=1}^N$ be the dataset consisting of N pairs of patches. We stack the vectors column-wise and obtain the following matrices: $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$.

We assume the following relationship between $\mathbf{x}_n$ and $\mathbf{z}_n$:

$$\mathbf{x}_n = \mathbf{f}(\mathbf{z}_n) + \boldsymbol{\varepsilon}_n \tag{1}$$

where $\boldsymbol{\varepsilon}_n$ is isotropic Gaussian noise with precision (inverse variance) β. As the simplest form of $\mathbf{f}$, we assume a *linear regression* model

$$\mathbf{f}(\mathbf{z}_n) = \mathbf{W}\mathbf{z}_n + \boldsymbol{\mu} \tag{2}$$

where $\mathbf{W}$ is a $D \times Q$ filtering matrix and $\boldsymbol{\mu}$ is a D-dimensional bias vector. Let $\mathbf{w}_d^{\mathrm{T}}$ be the dth row of $\mathbf{W}$. Since $\mathbf{w}_d$ is the filtering kernel for estimating the dth pixel of the high-resolution patch, $\mathbf{W}$ is regarded as a matrix built by stacking D filters. Even though linear regressors are simple, they are advantageous because they can be reduced to only the linear filtering of the given image that can be efficiently performed. This model leads to the probability distribution of $\mathbf{x}_n$

$$p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{W}, \boldsymbol{\mu}, \beta) = \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n + \boldsymbol{\mu}, \beta^{-1}\mathbf{I}_D) \tag{3}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian distribution (Appendix A) and $\mathbf{I}_D$ is the D-dimensional identity matrix.

In maximum-likelihood learning, the parameters are estimated by

$$(\mathbf{W}^*, \boldsymbol{\mu}^*) = \underset{(\mathbf{W},\boldsymbol{\mu})}{\arg\max} \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{W}, \boldsymbol{\mu}, \beta) \tag{4}$$

which can be performed easily; since this model is linear Gaussian, (4) is reduced to the least squares estimation. The optimal parameters are found as

$$\tilde{\mathbf{W}}^* = (\mathbf{X}\tilde{\mathbf{Z}}^{\mathrm{T}})(\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^{\mathrm{T}})^{-1} \tag{5}$$

where $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{Z}}$ are extended matrices to include $\boldsymbol{\mu}$ and defined as

$$\tilde{\mathbf{W}} = [\mathbf{W} \quad \boldsymbol{\mu}], \quad \tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z} \\ \mathbf{1}^{\mathrm{T}} \end{bmatrix}. \tag{6}$$

We refer to the filter trained by maximum likelihood as a maximum-likelihood expansion filter (MLEF). This training rule (5) can be considered as a special case of RS [1] with the number of components equated to one.

Given a new image to expand, the trained MLEF estimates high-resolution patches $\mathbf{x}$ from low-resolution patches $\mathbf{z}$ by the following filtering equation:

$$\mathbf{x} = \tilde{\mathbf{W}}^* \begin{bmatrix} \mathbf{z} \\ 1 \end{bmatrix} = \mathbf{W}^*\mathbf{z} + \boldsymbol{\mu}^*. \tag{7}$$

Note that maximum-likelihood estimation inherently suffers from overfitting; that is, an increase in the size of the filters beyond a certain complexity increases the generalization (test) error, although the training error always decreases [19].

A natural idea for preventing overfitting by support selection is to use sparse regularization on filter coefficients. Lasso [20] uses L_1 norm regularization on filter coefficients and efficient implementation called Lars [21] is available to draw the entire regularization path. We call the filter estimated by the Lasso L_1-regularized expansion filter (L1EF). Lasso regularization has never been performed in the context of image expansion

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

filter learning, and Section V would be the first report of how L1EF works in image expansion.

III. Sparse Bayesian Learning

In this section, we discuss sparse Bayesian estimation in the filtering model, which we call sparse Bayesian expansion filter (SBEF). According to the Bayesian methodology, all the parameters $(\mathbf{W}, \boldsymbol{\mu}, \beta)$ are treated as random variables, and *prior distributions* with parameters $\mathbf{A} = [\alpha_{dq}]$ and ρ are placed on them as follows:

$$p(\mathbf{W}|\mathbf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{N}(w_{dq}|0, \alpha_{dq}^{-1}) \tag{8}$$

$$p(\boldsymbol{\mu}|\rho) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{0}, \rho^{-1}\mathbf{I}_D) \tag{9}$$

$$p(\beta) = \mathcal{G}(\beta|a_{\beta 0}, b_{\beta 0}) \tag{10}$$

where $\mathcal{G}(\cdot)$ is the gamma distribution (Appendix A). We further place hierarchical priors on parameters $\mathbf{A}$ and ρ as

$$p(\mathbf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{G}(\alpha_{dq}|a_{\alpha 0}, b_{\alpha 0}) \tag{11}$$

$$p(\rho) = \mathcal{G}(\rho|a_{\rho 0}, b_{\rho 0}). \tag{12}$$

In the equations above, $a_{\bullet 0}$ and $b_{\bullet 0}$ are hyperparameters determined manually. The other variables are all determined automatically through Bayesian estimation. Note that the above distributions are all natural conjugate priors; thus, posterior distributions have the same function shapes as the priors.

The prior for filtering matrix $\mathbf{W}$ (8) is the key to the sparsity. This resembles the priors used in sparse Bayesian estimation [4]–[7] and is called *automatic relevance determination* (ARD), which was first introduced for neural networks [22]. Parameters α_{dq} work as regularizers that pull w_{dq} toward prior mean 0. Therefore, if the values of α_{dq} are very large, the estimated values of w_{dq} become very small. It is theoretically known [5], [23] that in this sparse Bayesian type of estimation, α_{dq}'s that satisfy a certain condition diverge to infinity; therefore, the corresponding elements of $\mathbf{W}$ become zero and hence are pruned from the filtering supports. In other words, the elements of $\mathbf{W}$ irrelevant to filtering are automatically switched off. The experiments in Section V will actually illustrate such behaviors. Since a detailed account of ARD is beyond the scope of this manuscript, see [4]–[7], [22], [23] for an in-depth discussion of ARD and sparse Bayesian estimation.

A graphical model representing the statistical dependency structure of SBEF is shown in Fig. 2. The joint probability is decomposed based on the model as

$$p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta, \mathbf{X}|\mathbf{Z}) = p(\mathbf{A})p(\mathbf{W}|\mathbf{A})p(\rho) \times p(\boldsymbol{\mu}|\rho)p(\beta)\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{W}, \boldsymbol{\mu}, \beta). \tag{13}$$

The filtering equation that maps a low-resolution patch $\mathbf{z}$ to a corresponding high-resolution patch $\mathbf{x}$ is simply given by the mean value of the predictive distribution

$$\mathbb{E}(\mathbf{x}) = \int \mathbf{x} p(\mathbf{x}|\mathbf{z}, \mathcal{D})\, d\mathbf{x}. \tag{14}$$

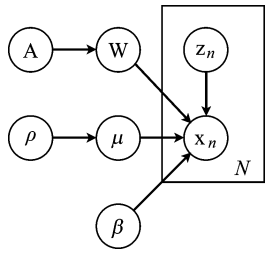


Fig. 2. Graphical model of sparse Bayesian expansion filters.

Predictive distribution $p(\mathbf{x}|\mathbf{z}, \mathcal{D})$ is calculated by

$$p(\mathbf{x}|\mathbf{z}, \mathcal{D}) = \int p(\mathbf{x}|\mathbf{z}, \mathbf{W}, \boldsymbol{\mu}, \beta) \times p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D})\, d\mathbf{A} d\mathbf{W} d\boldsymbol{\mu} d\rho d\beta \tag{15}$$

where $p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D})$ is the posterior distribution given data $\mathcal{D} = (\mathbf{X}, \mathbf{Z})$, derived by the Bayes theorem as

$$p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D}) = \frac{p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta, \mathbf{X}|\mathbf{Z})}{\int p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta, \mathbf{X}|\mathbf{Z})\, d\mathbf{A} d\mathbf{W} d\boldsymbol{\mu} d\rho d\beta}. \tag{16}$$

However, an analytical evaluation of the true predictive distribution is intractable because it is a complex of Gaussian and gamma variables. Therefore, we adopt an efficient computation procedure based on variational approximation, as described in the following section.

IV. Variational Inference

A. Variational Approximation

To overcome the intractability, posterior distribution $p(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta|\mathcal{D})$ is approximated by distribution $q(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta)$ that is restricted to a tractable class of distributions on which we impose a factorization property

$$q(\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta) = q(\mathbf{A})q(\mathbf{W})q(\rho)q(\boldsymbol{\mu})q(\beta). \tag{17}$$

We call q the trial distribution. Let the latent variables be denoted by $\boldsymbol{\eta} = \{\mathbf{A}, \mathbf{W}, \rho, \boldsymbol{\mu}, \beta\}$ for notational simplicity. Within the restricted distribution space, we search for the optimal trial distribution that minimizes Kullback–Leibler (KL) divergence to the true posterior distribution

$$q^{\star}(\boldsymbol{\eta}) = \arg\min_{q} D_{\mathrm{KL}}(q(\boldsymbol{\eta})||p(\boldsymbol{\eta}|\mathcal{D})) \tag{18}$$

where KL divergence is defined by

$$D_{\mathrm{KL}}(q(\boldsymbol{\eta})||p(\boldsymbol{\eta}|\mathcal{D})) = -\int q(\boldsymbol{\eta}) \ln \frac{p(\boldsymbol{\eta}|\mathcal{D})}{q(\boldsymbol{\eta})}\, d\boldsymbol{\eta} \tag{19}$$

$$= -\left\langle \ln \frac{p(\boldsymbol{\eta}|\mathcal{D})}{q(\boldsymbol{\eta})} \right\rangle_{\boldsymbol{\eta}}. \tag{20}$$

Here, $\langle\cdot\rangle_{\boldsymbol{\eta}}$ is the expectation operator with respect to $q(\boldsymbol{\eta})$. KL divergence is similar to the distance because it is nonnegative, i.e., $D_{\mathrm{KL}}(q||p) \geq 0$, for any q and p, and $D_{\mathrm{KL}}(q||p) = 0$ if and only if q and p are equivalent distributions.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

This variational optimization problem can be analytically solved if we optimize only one factor $q(\boldsymbol{\eta}_i)$, fixing the other factors $q(\boldsymbol{\eta}_{\setminus i}) = \prod_{j\neq i} q(\boldsymbol{\eta}_j)$ [24], [25]

$$q^*(\boldsymbol{\eta}_i) = \frac{\exp\{-\langle \ln p(\boldsymbol{\eta}, \mathbf{X}|\mathbf{Z})\rangle_{\boldsymbol{\eta}_{\setminus i}}\}}{\int \exp\{-\langle \ln p(\boldsymbol{\eta}, \mathbf{X}|\mathbf{Z})\rangle_{\boldsymbol{\eta}_{\setminus i}}\}\, d\boldsymbol{\eta}_i}. \tag{21}$$

In the following subsection, we present the optimal factors $q^*(\mathbf{A})$, $q^*(\mathbf{W})$, $q^*(\rho)$, $q^*(\boldsymbol{\mu})$, and $q^*(\beta)$. To find joint minimum $q^\star$, we iterate factor-wise, or coordinate-descent, optimization until convergence. Variational estimation is becoming popular in the field of signal processing as can be seen in a recent tutorial article [25].

B. Optimal Trial Distribution

Optimal distribution is sought by iterating the computation of the following optimal trial factors:

$$q^*(\mathbf{A}) = \prod_{d=1}^{D}\prod_{q=1}^{Q} \mathcal{G}(\alpha_{dq}|a_{\alpha dq}, b_{\alpha dq}) \tag{22}$$

$$q^*(\mathbf{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d|\mathbf{m}_{\mathrm{w}}^{(d)}, \boldsymbol{\Sigma}_{\mathrm{w}}^{(d)}) \tag{23}$$

$$q^*(\rho) = \mathcal{G}(\rho|a_\rho, b_\rho) \tag{24}$$

$$q^*(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}_\mu, \boldsymbol{\Sigma}_\mu) \tag{25}$$

$$q^*(\beta) = \mathcal{G}(\beta|a_\beta, b_\beta) \tag{26}$$

where the parameters are given by

$$a_{\alpha dq} = a_{\alpha 0} + \frac{1}{2}, \quad b_{\alpha dq} = b_{\alpha 0} + \frac{1}{2}\langle w_{dq}^2\rangle \tag{27}$$

$$\boldsymbol{\Sigma}_{\mathrm{w}}^{(d)} = \left(\langle \mathrm{diag}(\alpha_{d1}, \ldots, \alpha_{dQ})\rangle + \langle\beta\rangle \sum_{n=1}^{N} \mathbf{z}_t \mathbf{z}_t^{\mathrm{T}}\right)^{-1} \tag{28}$$

$$\mathbf{m}_{\mathrm{w}}^{(d)} = \langle\beta\rangle \boldsymbol{\Sigma}_{\mathrm{w}}^{(d)} \sum_{n=1}^{N} (x_{dn} - \langle\mu_d\rangle)\mathbf{z}_n \tag{29}$$

$$a_\rho = a_{\rho 0} + \frac{D}{2}, \quad b_\rho = b_{\rho 0} + \frac{1}{2}\langle \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}\rangle \tag{30}$$

$$\boldsymbol{\Sigma}_\mu = \frac{1}{\langle\rho\rangle + N\langle\beta\rangle}\mathbf{I}_D \tag{31}$$

$$\mathbf{m}_\mu = \langle\beta\rangle \boldsymbol{\Sigma}_\mu \sum_{n=1}^{N} (\mathbf{x}_n - \langle\mathbf{W}\rangle\mathbf{z}_n) \tag{32}$$

$$a_\beta = a_{\beta 0} + \frac{ND}{2} \tag{33}$$

$$b_\beta = b_{\beta 0} + \frac{1}{2}\sum_{n=1}^{N}\{\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n - 2\mathbf{x}_n\langle\mathbf{W}\rangle\mathbf{z} - 2\mathbf{x}_n^{\mathrm{T}}\langle\boldsymbol{\mu}\rangle + \mathbf{z}_n^{\mathrm{T}}\langle\mathbf{W}^{\mathrm{T}}\mathbf{W}\rangle\mathbf{z}_n^{\mathrm{T}} + 2\mathbf{z}_n^{\mathrm{T}}\langle\mathbf{W}\rangle^{\mathrm{T}}\langle\boldsymbol{\mu}\rangle + \langle\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu}\rangle\}. \tag{34}$$

The expectations remaining in the above equations can be evaluated easily by using well-known results in statistics (Appendix A). Distributions (22)–(26) have the same function shapes as their priors due to the natural conjugate prior setting. In (22) and (23), further independence, which was not assumed in (17), is automatically derived from the model structure. These optimal factors are mostly the same as those derived in [6] as relevance vector machine regression; the differences are the existence of the bias term ($\boldsymbol{\mu}$) and the disuse of kernel functions.

Input: High- and low-resolution patch pairs $\mathcal{D} = (\mathbf{X}, \mathbf{Z})$ and sparsity hyperparameter $a_{\alpha 0}$.
Output: Filter parameters $\mathbf{M}_{\mathrm{W}}$ and $\mathbf{m}_\mu$.

1: **repeat**
2: Update $q(\mathbf{A})$ by computing $a_{\alpha dq}$ and $b_{\alpha dq}$ using (26).
3: Update $q(\mathbf{W}) = \prod_{d=1}^{D} q(\mathbf{w}_d)$ by computing $\boldsymbol{\Sigma}_{\mathrm{w}}^{(d)}$ and $\mathbf{m}_{\mathrm{w}}^{(d)}$ using (27), (28).
4: Update $q(\rho)$ by computing a_ρ and a_ρ using (29).
5: Update $q(\boldsymbol{\mu})$ by computing $\boldsymbol{\Sigma}_\mu$ and $\mathbf{m}_\mu$ using (30), (31).
6: Update $q(\beta)$ by computing a_β and b_β using (32), (33).
7: Set $\langle\alpha_{dq}\rangle$ to infinity if it is above e^{20}.
8: **until** Relative change of $\mathbf{M}_{\mathrm{W}}$ is sufficiently small.

Fig. 3. Training algorithm for SBEF.

We denote the mean of joint trial distribution $q(\mathbf{W})$ by $\mathbf{M}_{\mathrm{W}}$; that is, we put $\mathbf{M}_{\mathrm{W}} = [\mathbf{m}_{\mathrm{w}}^{(1)}, \ldots, \mathbf{m}_{\mathrm{w}}^{(D)}]^{\mathrm{T}}$. The filtering equation for variational SBEF image expansion is obtained by substituting the true posterior distribution with the trial distribution, which results in

$$\mathbb{E}(\mathbf{x}) \approx \langle\mathbf{x}\rangle = \langle\mathbf{W}\rangle\mathbf{z} + \langle\boldsymbol{\mu}\rangle = \mathbf{M}_{\mathrm{W}}\mathbf{z} + \mathbf{m}_\mu. \tag{35}$$

This is the expansion equation for SBEF. When we expand a given image, this linear filtering (35) is repeated for every low-resolution patch.

C. SBEF Learning Algorithm

1) Training Procedure: An algorithmic procedure to train an SBEF is shown in Fig. 3. As a criterion to check convergence and stop iterating (22)–(26), we monitor the relative change of the Frobenius norm of $\mathbf{M}_{\mathrm{W}}$

$$\Delta = \frac{||\mathbf{M}'_{\mathrm{W}} - \mathbf{M}_{\mathrm{W}}||_{\mathrm{F}}}{||\mathbf{M}'_{\mathrm{W}}||_{\mathrm{F}}} \tag{36}$$

where $\mathbf{M}'_{\mathrm{W}}$ is the matrix at the previous iteration step; the algorithm is terminated when $\Delta < 10^{-6}$. To accelerate the convergence, the expected values of α_{dq} are thresholded and set to infinity when they are greater than threshold e^{20}. We conducted several preliminary experiments and confirmed that if α_{dq} converge, their converged values never exceed e^{15}; therefore, this threshold of e^{20} is sufficiently large.

Hyperparameters $a_{\bullet 0}$, $b_{\bullet 0}$ must be determined manually. We used a hyperparameter setting of noninformative limit $a_{\beta 0} = b_{\beta 0} = a_{\rho 0} = b_{\rho 0} = 0$ for β and ρ. For α, we assume $b_{\alpha 0} = 0$ but $a_{\alpha 0} \neq 0$; this allows us to control the value of α_{dq} to facilitate the divergence of $\langle\alpha_{dq}\rangle$. We call $a_{\alpha 0}$ the *sparsity hyperparameter* since this value determines the degree of sparseness, as will be seen in the experiments. Although having zero hyperparameters makes the priors improper, this is not a problem since the posterior computation of $a_\bullet$, $b_\bullet$ is well defined.

2) Color Image Expansion: Color consideration is important for real applications. There are at least three methods for color image expansion.

1) Expand each RGB component separately.
2) Learn the direct relationship between low- and high-resolution *color* patches. In other words, extend $\mathbf{x}$ and $\mathbf{z}$ to include the three color components.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

3) Convert the color space from RGB to YIQ, expand only the luminance component, and then convert it back to the RGB space.

Although 2) should yield the best quality, we adopt 3) for efficient computations in the experiments.

3) Symmetry in Filter Supports: Since natural image patches harbor symmetry, we expect filters $\mathbf{w}_d$ to have symmetric supports. This notion can be easily integrated into SBEF by constraining α_{dq} to be symmetric over different d. We enforce symmetry in both horizontal and vertical directions; that is, the support of the filter for estimating the value of the top-left pixel is horizontally symmetric with that of the filter for the top-right pixel, vertically symmetric with that for the bottom-left pixel, and so on.

V. Experimental Results

We conducted five experiments. In the first experiment, we observed what support would be selected by SBEF, MLEF, and L1EF and compared the performances of them. In the second, we saw the effect of mismatch in anti-aliasing smoothing between training/test datasets. Third, we visually compared SBEF with the example-based superresolution method by Freeman *et al.* [2]. The fourth experiment demonstrated how the performance of Atkins' RS was affected by the support discovered by our SBEF method. The final experiment was to see when learned filters fail to surpass simple interpolation approaches.

Since learned filters are optimized for the training dataset, we must clearly separate training and testing datasets. When assessing the quality of the expanded image, we only measured the peak signal-to-noise ratio (PSNR) between expanded image $\hat{\boldsymbol{\xi}}$ and original image $\boldsymbol{\xi}^*$ for the luminance component. PSNR is defined by

$$\mathrm{PSNR}(\boldsymbol{\xi}^*, \hat{\boldsymbol{\xi}}) = 10 \log_{10} \frac{\kappa^2}{\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\xi}}\|^2/(r^2 MN)} \quad \text{(dB)} \quad (37)$$

where κ is the maximum pixel value and r^2MN is the number of pixels. All the PSNR values presented in the experiments are test (or generalization) PSNRs; i.e., they are measured for images *not included* in the training dataset. Since the main focus of this study is to offer a good tradeoff between high quality (high PSNR) and low computational cost (small support), we must also heed the size of the support of the learned filters.

Training and test datasets were generated by the following procedures. The pixel values are first converted into double-precision floating points within $[0, 1]$ and transformed to luminance values if the original image has color channels. High-resolution patches are prepared by cutting them into non-overlapping pieces. To make low-resolution patches, first the high-resolution images are blurred by an anti-aliasing filter and subsampled by specified factor r to extract overlapping patches of size $m \times m$. For the training datasets, low-resolution patches stemming from the boundaries are discarded, and the corresponding high-resolution patches are not used. For test datasets, to extract patches near the boundaries, the low-resolution image is extended by pixel replication.

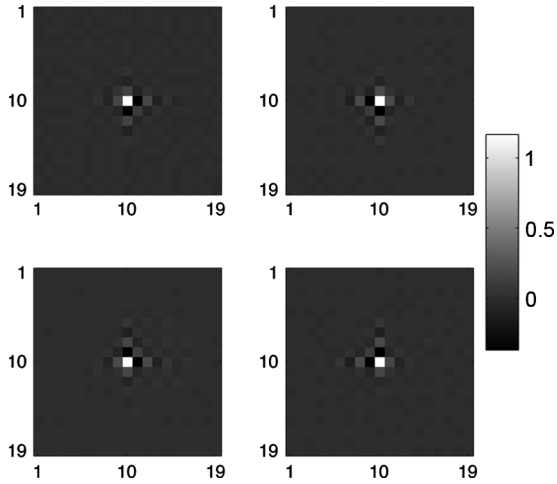


Fig. 4. Learned MLEF filters. All coefficients are nonzero.

A. Comparison of SBEF, MLEF, and L1EF

The expanding factor is chosen to be $r = 2$. The training dataset is produced from eight images of size 256×256 (#4.1.[01-08] in the USC-SIPI image database [26]), resulting in a total of $N = 8 \cdot 256^2/r^2 = 131,072$ patch pairs. We used seven images (#4.2.[01-07] in [26]) as test images on which PSNRs are measured, and the mean PSNR is used to assess the quality of the learned filters. A cubic kernel is used for anti-aliasing blurring both on the training and test images. The Lars–Lasso algorithm [21] is used to obtain L1EFs.

The MLEF performance was measured with low-resolution patches whose size varied from $3 \times 3 = 9$ to $19 \times 19 = 361$. Fig. 4 shows the filters of the trained MLEF when the size of the low-resolution patches is 19×19. There is no nonzero element in the filter coefficients; thus, the size of the support is 19×19. Although skewed, the coefficients have oscillation and look somewhat similar to the sinc function, as expected from the results of Triggs [8]. The maximum mean PSNR of 31.83 dB is attained when the patch size is $11 \times 11 = 121$, and the use of larger patches only degrades the performance, showing typical overfitting.

The SBEF training algorithm was executed with a fixed size of low-resolution patches, 19×19, by varying hyperparameters $a_{\alpha 0}$ from 1 to 210. The shapes of the learned SBEF filters $\mathbf{m}_{\mathrm{w}}^{(d)}$ with $a_{\alpha 0} = 20$ are shown in Fig. 5, and the supports (regions where the filters have nonzero values) are shown in Fig. 6. The effective sizes of the learned supports are all 29. Compared to the size of the best MLEF, the size of the SBEF filters is reduced by $11^2 - 29 = 92$ pixels for each of the four filters; however, the PSNR remained the same. The same effective supports were learned with patches larger than 11×11 since the pixels of the outer region were automatically pruned. We note that SBEF filters learned without the symmetricity constraint on α_{dq} had similar supports, but with slightly (below 0.1 dB) worse PSNRs.

An interesting point is that the learned supports shown in Fig. 6 have irregular shapes. From the shapes of the learned supports, the direct horizontal and vertical pixels are highly relevant for estimating high-resolution pixels, but the diagonal pixels are of less importance. The filters bulge toward the center; the filter

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

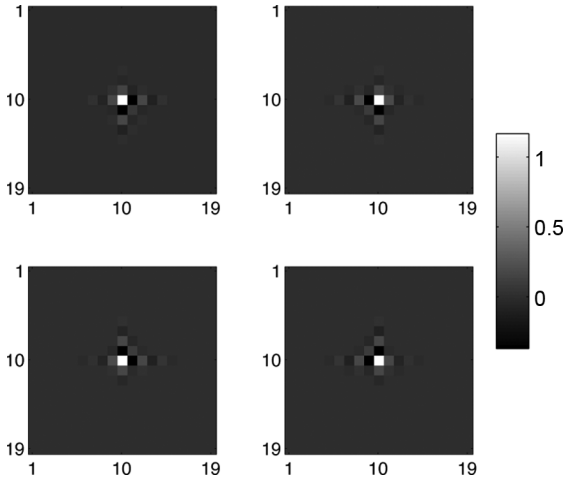


Fig. 5. Learned SBEF filters. They resemble those of MLEF, but the marginal coefficients are exactly zero, as suggested by the supports in Fig. 6.

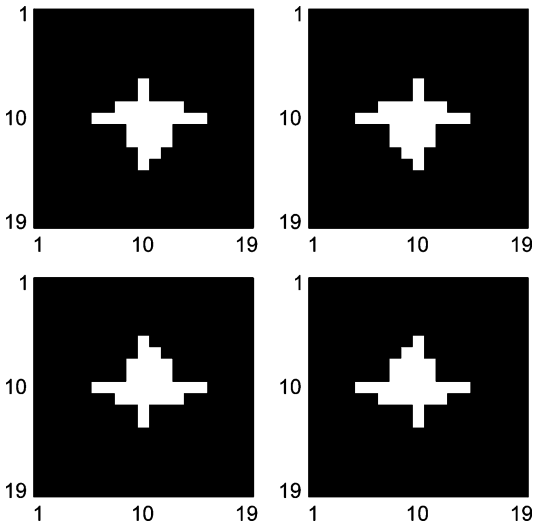


Fig. 6. Supports of learned SBEF filters.

for estimating the top-left high-resolution pixels is swelled toward the bottom right, the top-right filter toward the bottom left, and so on. These irregular shapes could be attributed to the square shape of pixels; the irregularity would not be observed if pixels were arranged as a honeycomb grid.

The low-resolution training patches of size 19×19 were also given to the Lars–Lasso algorithm for training L1EF, which found sparse supports whose sizes ranging from 0 to 19^2. Fig. 7 shows the shapes of the L1EF support when the mean support size is 24. Although around the central regions the supports appear to have similar shapes to those of SBEF, the supports are somehow messy so that some coefficients are scattered on the marginal regions. Coefficients of the central regions had oscillations similar to the other filters (figure not shown for saving space).

Table I shows the effective support sizes and mean PSNRs for the SBEF, MLEF, and bicubic methods. Fig. 8 plots the content of Table I together with the performance of the L1EF; the crosses, circles, and dots indicate the PSNRs of the SBEF, MLEF, and L1EF, respectively. The learned filters have significantly better PSNRs than the bicubic method. For support sizes

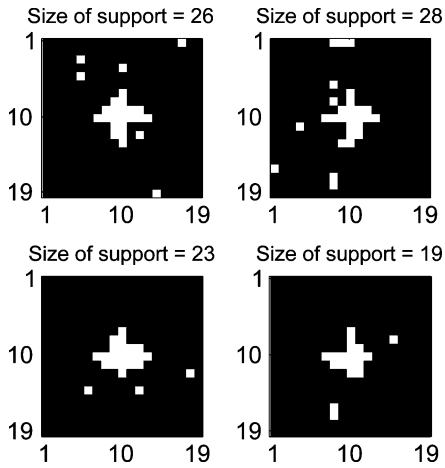


Fig. 7. Supports of learned L1EF filters when the mean support size of the four filters is 24.

TABLE I
EFFECTIVE SUPPORT SIZES LEARNED AND MEAN PSNRS FOR SEVEN TEST IMAGES

SBEF			MLEF	
$a_{\alpha 0}$	Support	Mean PSNR	Support	Mean PSNR
1	361	31.82	361	31.81
2	139	31.83	289	31.82
3	76	31.83	225	31.82
4	48	31.83	169	31.82
5	44	31.83	121	31.83
10	37	31.83	81	31.82
20	29	31.83	49	31.80
30	23	31.82	25	31.71
50	20	31.80	9	31.33
90	15	31.76		
130	12	31.70	Bicubic Interpolation	
170	9	31.63	Support	Mean PSNR
210	8	31.59	16	30.89

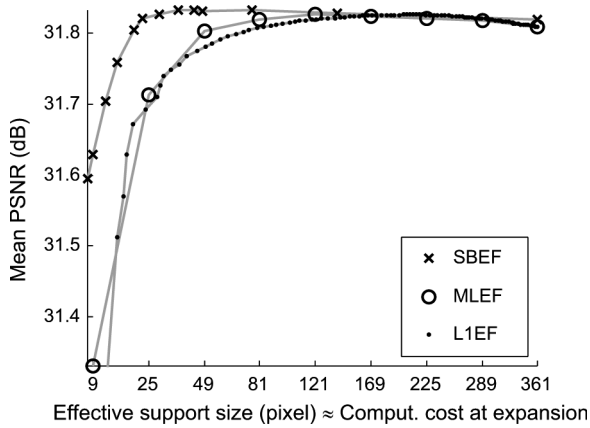


Fig. 8. Generalization performance of expansion filters: mean PSNRs for seven test images versus effective sizes of support. Crosses, circles, and dots show the performance of SBEF, MLEF, and L1EF, respectively.

smaller than 25, SBEF yields PSNRs that are higher by 0.2–0.3 dB than those of MLEF and L1EF. Remember that the support size should be controlled directly for MLEF, whereas for SBEF the original patch size is fixed at 19×19 and sparsity hyperparameter $a_{\alpha 0}$ is controlled instead. Increasing the low-resolution patch size does not affect the acquired effective support size for SBEF, but the support size of MLEF increases endlessly with the patch size; this implies that, for MLEF, there is no criterion

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

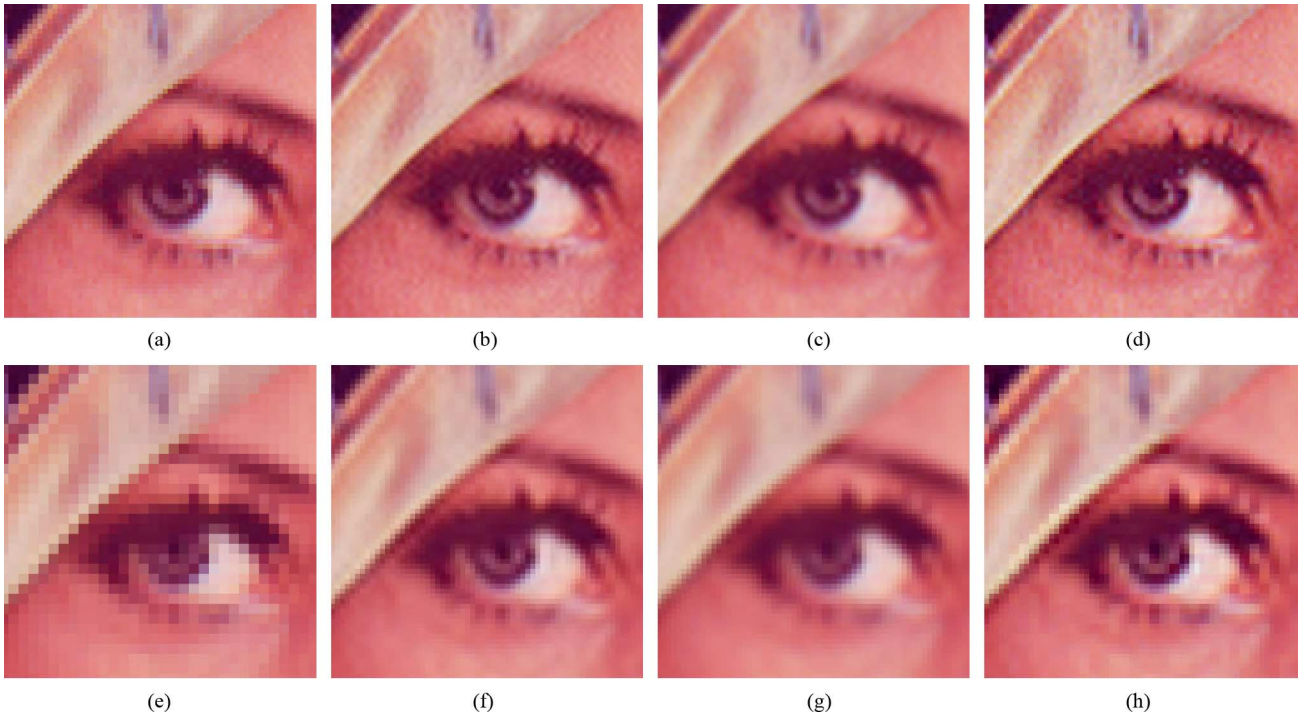


Fig. 9. Expansion results for Lena image. (a) Original image (#4.2.04 in [26]), whose expansions are (b)–(d). (e) Low-resolution image [(a) is blurred by cubic kernel and subsampled], whose expansions are (f)–(h). PSNR values for (f)–(h) are calculated only for the shown region. (a) Original image. (b) SBEF expansion of (a). (c) Cubic expansion of (a). (d) Sharpening of (c). (e) Low-resolution image (f). SBEF expansion of (e); 35.59 dB. (g) Cubic expansion of (e); 33.05 dB. (h) Sharpening of (g); 30.95 dB.

to automatically choose the optimal support size and shape. Although L1EF is able to acquire sparseness, it has unwanted coefficients distant from the center; these unwanted coefficients lessen the performance of L1EF over SBEF when compared at the same-size points. These results suggest the advantage of SBEF over MLEF and L1EF because SBEF achieved higher PSNRs with smaller flexible supports, which results in relatively higher performance with smaller computational costs when expanding the images.

Here, we would like to point out that simple support selection methods for MLEF do not work well. The minimum of the absolute value of the trained filter coefficients shown in Fig. 4 is as small as 10^{-6} compared to the mean absolute value of 2×10^{-2}. Then one might consider cutting off the coefficients smaller than a specified threshold. However, reducing the support size by this simple cutoff method only decreased PSNR; if the performance were plotted in Fig. 8, it would always be below the line connecting the circles. Cross validation is another possible method to find the optimal support, that is, a comparison of all 2^{m^2} combinations of the shapes of support. The computational cost is obviously prohibitive for a large m and thus it is less realistic.

As another performance reference, we tested the performance of the reconstruction-based image superresolution method that employs total-variation (TV) regularization [18]. The TV method was modified to use only one image, and the regularization hyperparameter was hand-tuned for each test image to produce the highest PSNR. The mean PSNR for the seven test images was 31.11 dB, which was higher than that of the bicubic method but lower than those of the learned filters. Almost the same mean test PSNR was obtained when using the L_2 regularization as in [17].

Before closing the first experiment, we show the image expansion results comparing SBEF with hyperparameter $a_0 = 20$, bicubic interpolation, and bicubic interpolation + post sharpening (the "Sharpen More" filter of Adobe Photoshop CS). The expansion results and the source images given to the expanders are shown in Fig. 9.

B. *Effect of Anti-Aliasing Mismatch*

It is plausible that the SBEF performance depends on the anti-aliasing blurring kernel that was used to generate the training data, and that has really blurred the given image to expand. Then we conducted the second experiment to see how the mismatch in the anti-aliasing kernels influences the expansion performance. The original images for training and testing are the same as those in the previous subsection, which were smoothed using the five kernels shown in Fig. 10.

We trained SBEF with sparsity hyperparameter $a_0 = 20$ and measured the test PSNRs. See Table II for the performance under the training/test kernel mismatch. There is a tendency that high PSNRs are obtained when the same kernel is used for both training and testing, while mismatch in the anti-aliasing kernels does not always cause serious deterioration of the expansion quality. Whatever kernel is used for training, on the other hand, low PSNRs are obtained for the delta-smoothed test images; this shows agreement with the findings of Triggs [8], who reported that good expansion is possible only when good anti-aliasing

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

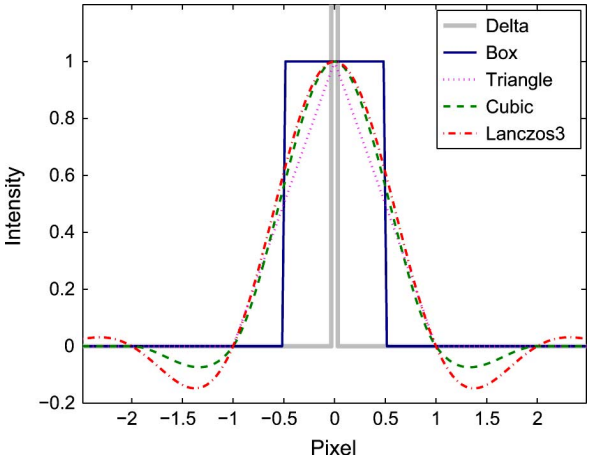


Fig. 10. Anti-aliasing kernels.

TABLE II
MEAN PSNRS FOR SEVEN TEST IMAGES UNDER TRAINING/TEST ANTI-ALIASING KERNEL MISMATCH

		Test kernel					Mean
		Delta	Box	Triangle	Cubic	Lanczos3	
Training kernel	Delta	30.81	**31.44**	30.69	**31.48**	**31.60**	31.20
	Box	30.84	**31.54**	30.91	**31.65**	**31.67**	31.32
	Triangle	28.47	29.27	**31.77**	30.64	29.61	29.95
	Cubic	30.48	31.23	31.15	**31.83**	**31.70**	31.28
	Lanczos3	30.61	31.31	30.76	**31.71**	**31.90**	31.26
	Mean	30.24	30.96	31.06	31.46	31.30	

(e.g., sinc) is used for images to expand. Similar results were also observed for the MLEF filters.

C. Comparison With Example-Based Superresolution

We compared our expansion method to one of the learning-based superresolution algorithms, the example-based superresolution of Freeman *et al.* [2]. We trained the SBEF with $a_0 = 20$ using all the original images used for generating training and test datasets in Section V-A, anti-aliased by the cubic kernel. The results are shown in Fig. 11. Fig. 11(a) and (b) is taken from [27].[1] SBEF produces a sharper result than the bicubic interpolation. Although quantitative evaluation is not provided since the ground truth is unknown, the estimate of the example-based method looks better than those of the other methods. This performance gap may be explained by the complexity of the example-based method. We should be aware that the aims of SBEF and the example-based superresolution are different; the former attempts to obtain compact linear filters for efficient image processing, whereas the latter's efforts are devoted to developing high-performance (and high-cost) machinery by searching in a large example database.

D. Using SBEF's Support for Atkins' RS

In this experiment, we demonstrate how the performance of Atkins' RS is affected by integrating the supports found by SBEF. The following two support settings were used: the 5×5 support, which is the same as proposed in the original RS [1] and the one shown in Fig. 6. In the EM clustering phase, fully parametrized covariance matrices were estimated, although the original RS uses isotropic covariance, because the former setting produced higher PSNRs. Since RS depends on the initialization of the EM algorithm, we used 12 different initializations and measured the means and standard deviations of the PSNRs of the resultant 12 RS interpolators. The training and test datasets are the same as in Section V-A. The number of mixture components varied from 10 to 120.

[1]Free copying for research purposes is permitted by the copyright holder, MERL.

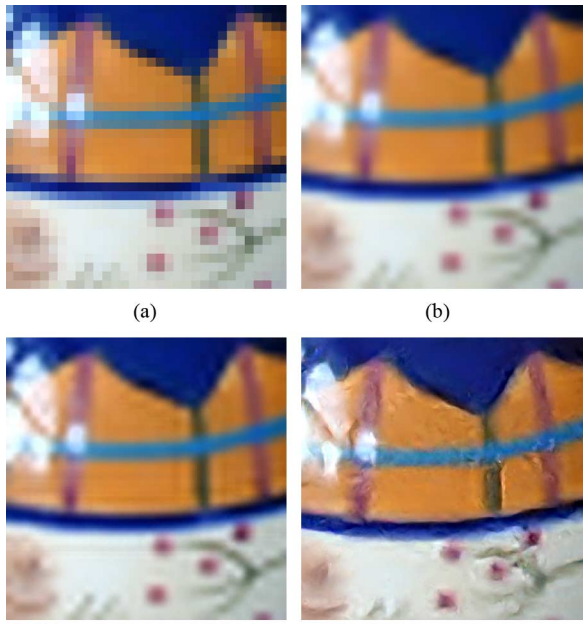


Fig. 11. Comparison with example-based superresolution method. (a) Low-resolution. © MERL [27]. (b) Bicubic interpolation. (c) SBEF expansion. (d) Example-based. © MERL [27].

Fig. 12 shows the results of RS image expansion with 5×5 and SBEF support settings and the performance of the best SBEF obtained in Section V-A. Surprisingly, the mean PSNRs of the original RS are not higher than those by the non-mixture SBEF. The RS with the SBEF-based support exhibits improvement in PSNR over the sole SBEF when the number of mixture components is less than 30.

E. Failure of Learned Filters in Severe Situations

In this final experiment, filters were learned in challenging situations where the noise levels were high or the magnification factor was large. The training and test datasets are the same as in Section V-A and the cubic kernel is used for all anti-aliasing tasks.

So far all the experiments were performed without adding noise to low-resolution images. Here we see how the expansion performance is affected by the presence of noise. The magnification factor is 2. White Gaussian noise of SNR varied from 10 to 40 dB are added to both the training and test low-resolution images, and Fig. 13 shows the performance of the SBEF trained with $a_0 = 20$, cubic interpolation, pre-denoising + cubic interpolation, and cubic interpolation + post-denoising. As the denoising algorithm, we used the BLS-GSM Image Denoising Toolbox [28]. When the noise strength is low (40 dB), the SBEF's mean test PSNR is approximately 0.9 dB higher than those of the other algorithms. However, as the noise level

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

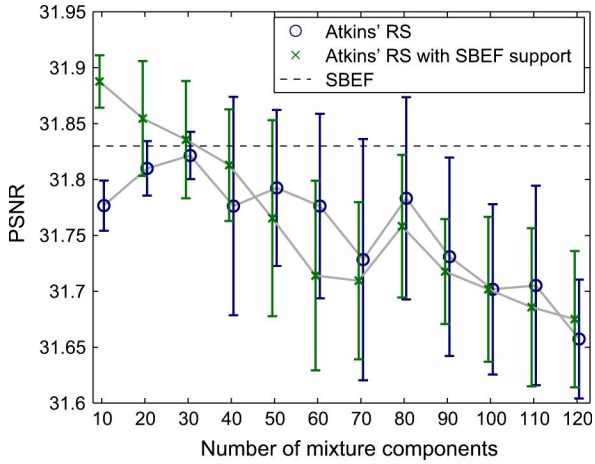


Fig. 12. Test PSNRs of RS methods: mean ± standard deviation for 12 initializations of clustering.

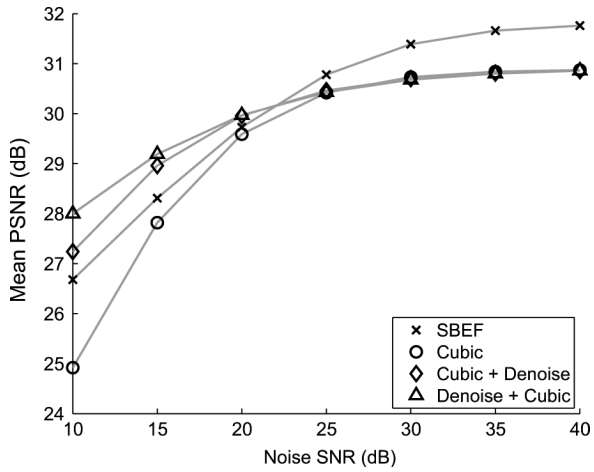


Fig. 13. Mean test PSNRs under different noise levels.

becomes severe toward 10 dB, the PSNR values decline and the SBEF performance drops below that of the pre-denoising + cubic algorithm.

Next we see how the performance changes as the magnification factor is increased. The magnification factors up to 14 are used. Fig. 14 shows the performance of the SBEF trained with $a_0 = 0$ and cubic interpolator. When the factor is 2, the SBEF outperforms the cubic by the same amount as shown in Table I. However, as the magnification factor gets larger, their mean test PSNR values approach each other and the advantage of SBEF becomes hard to be seen.

VI. Discussion on Modeling Direction

Our framework for estimating high-resolution images has an interesting relationship with variational Bayesian super-resolution methods [17], [18]. Essentially, both frameworks approach the same quantity from different directions; the same goal of Bayesian estimation is the probability distribution of the high-resolution image conditioned on the low-resolution observations, $p(\tilde{\mathbf{x}}|\tilde{\mathbf{z}})$, where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ are the entire (non-patch) high-resolution and low-resolution images, respectively. For

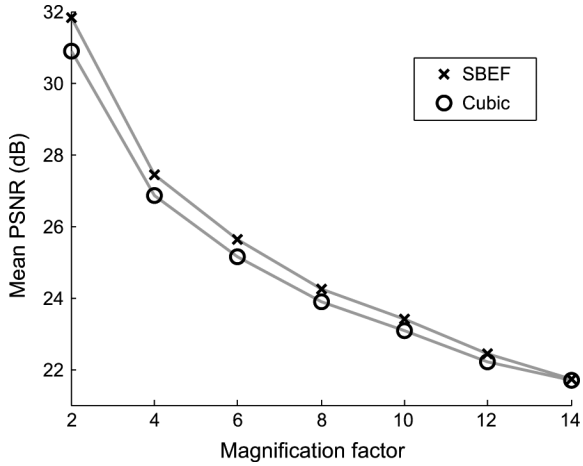


Fig. 14. Mean test PSNRs under different magnification factors.

simplicity, in this section, we ignore other variables than the images without loss of generality.

The authors of [17] and [18] start by defining a prior probability on high-resolution images $p(\tilde{\mathbf{x}})$ and the probability of observing low-resolution images $p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})$ that represents a physical observation process consisting of warping, blurring, downsampling, and noise addition. Then the distribution in demand is calculated by the Bayes theorem by

$$p(\tilde{\mathbf{x}}|\tilde{\mathbf{z}}) = \frac{p(\tilde{\mathbf{x}})p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})}{\int p(\tilde{\mathbf{x}})p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})\,d\tilde{\mathbf{x}}}. \tag{38}$$

They apply the variational Bayesian technique and derive iterative update equations for approximate inference. The resulting updating rule for the estimate of the high-resolution image is derived as a *linear* transform of the observations [17], [18].

On the other hand, our approach attempts to directly model $p(\mathbf{x}|\mathbf{z})$ by a linear model (3), which results in a local linear operation (filtering) given by (35) for estimating high-resolution images. The consideration of patch-wise distributions is equivalent to assuming patch-wise independence on the entire-image distribution.

This difference in the modeling direction is the same as the difference between "generative" and "discriminative" models for classification discussed in the machine learning community [29].

Although the two frameworks both result in linear operation for inversion, their inverse operators have different features coming from different parametrizations. In the "generative" superresolution framework by [17] and [18], we have to choose the observation model (e.g., the shape of the blur) and the prior model (it is usually difficult to incorporate the exact prior knowledge of the image; no one has successfully found *the true* distribution of natural images). Moreover, the generative framework requires accurate registration between the observed frames. The inverse filter is then constructed to correspond to the choice of the two models. Therefore, assuming that the models and registration are correct, the estimated images should be close to the ground truth. By contrast, in our framework, which we call "synthetical" since there is nothing to discriminate, the inverse operator is directly learned from a dataset, rather than defining

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

an explicit physical model to invert. Therefore, it depends on the characteristics (e.g., sampling distance, blurring kernels, scenes, or objects in images) of the training dataset, while it is advantageous because it skips the difficulties in choosing the precise prior and observation models. Nevertheless, the learned filters exhibit good generalization on test images unseen in the training data and differently anti-aliased images.

VII. Conclusion

We proposed SBEF, a synthetical model for acquiring compact yet high-performance image expansion filters based on sparse Bayesian estimation, and derived an efficient learning procedure for its parameters on the basis of variational approximation. We demonstrated that the compact filter supports, relevant to high-resolution image estimation, can be automatically selected by SBEF. Although the learning algorithm of SBEF described in Fig. 3 requires higher computational loads than the one-shot calculation (5) of MLEF, the supports of the filters learned by SBEF were significantly smaller than those learned by MLEF, and the PSNRs were higher even with the reduced numbers of the low-resolution pixels used for image expansion. Another sparse learning method for image expansion filters, L1EF, was designed using the Lasso; however, their sparse supports are somewhat scattered and do not contribute to improving the performance. These results are interesting from both the theoretical and technological aspects. Theoretically, they signify the discovery of an effective subspace where the optimal image regressors reside. Technically, they show that SBEF can be efficiently implemented by using only the relevant supports, which would be particularly beneficial when considering realistic applications to graphics software and various embedded systems, for example.

Finally we should be aware that, if we ignore the computational costs, the proposed linear regression framework is not competitive with nonlinear algorithms, including RS and example-based superresolution; it is only a first-order approximation to the nonlinear real world. Moreover, when the magnification factor is large or noise is severe, learned linear filters would not give the best performance. When plenty of computational resources is available, or when the observation process is too severe to recover by mere linear filtering, the complicated image expansion methods will be preferred.

Appendix
Statistical Distributions

In this section, we have listed the statistical distributions used in the text. The expectation with respect to each distribution is denoted by $\langle\cdot\rangle$.

- Gaussian distribution:

$$\mathcal{N}(\mathbf{x}|\mathbf{m},\boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} e^{-1/2(\mathbf{x}-\mathbf{m})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{m})} \tag{39}$$

$$\langle\mathbf{x}\rangle = \mathbf{m}, \quad \langle\mathbf{x}^{\mathrm{T}}\mathbf{x}\rangle = \mathbf{m}^{\mathrm{T}}\mathbf{m} + \mathrm{tr}(\boldsymbol{\Sigma}),$$
$$\langle\mathbf{x}\mathbf{x}^{\mathrm{T}}\rangle = \mathbf{m}\mathbf{m}^{\mathrm{T}} + \boldsymbol{\Sigma} \tag{40}$$

- Gamma distribution:

$$\mathcal{G}(\tau|a,b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} e^{-b\tau} \tag{41}$$

$$\langle\tau\rangle = \frac{a}{b}. \tag{42}$$

Acknowledgment

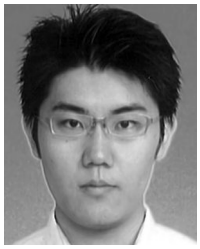
The authors would like to thank Dr. S. Oba at Kyoto University for his insightful comments on ARD and his careful reading of the early versions of the manuscript. They would also like to thank the anonymous reviewers for valuable comments, especially for pointing out the work of Triggs [8], for asking the comparison with the Lasso, for useful suggestions on experiments, and for triggering the discussion presented in Section VI. Fig. 11(a) and (b) is courtesy of the authors of [27] and MERL.

References

[1] C. B. Atkins, "Classification-based methods in optimal image interpolation," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1998.
[2] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
[3] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, Sep. 2002.
[4] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.
[5] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems (NIPS) 14*. Cambridge, MA: MIT Press, 2002, pp. 383–389.
[6] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Uncertainty in Artificial Intelligence (UAI)*. San Francisco, CA: Morgan Kaufmann, 2000, pp. 46–53.
[7] C. M. Bishop, "Bayesian PCA," in *Advances in Neural Information Processing Systems (NIPS) 11*. Cambridge, MA: MIT Press, 1999, pp. 382–388.
[8] B. Triggs, "Empirical filter estimation for subpixel interpolation and matching," in *Proc. Int. Conf. Computer Vision (ICCV)*, Vancouver, BC, Canada, Jul. 2001, vol. 2, pp. 550–557.
[9] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.
[10] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Tree-based resolution synthesis," in *Proc. Image Process., Image Quality, Image Capture Syst. (PICS) Conf.*, Savannah, GA, Apr. 1999, pp. 405–410.
[11] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *Proc. Int. Conf. Image Process. (ICIP)*, Thessaloniki, Greece, Oct. 2001, vol. 3, pp. 864–867.
[12] R. Yoakeim and D. Taubman, "Quantitative analysis of resolution synthesis," in *Int. Conf. Image Processing (ICIP)*, Singapore, Oct. 2004, vol. 3, pp. 1645–1648.
[13] R. Yoakeim and D. Taubman, "Interpolation specific resolution synthesis," in *Proc. Int. Conf. Image Process. (ICIP)*, San Antonio, TX, Sep. 2007, vol. 4, pp. IV-201–204.
[14] T. Akgun and Y. Altunbasak, "A coupled feature-filter clustering scheme for resolution synthesis," in *Int. Conf. Image Process. (ICIP)*, San Antonio, TX, Sep. 2007, vol. 5, pp. V-405–408.
[15] H. Siddiqui and C. A. Bouman, "Training-based descreening," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 789–802, Mar. 2007.
[16] A. Katsaggelos, R. Molina, and J. Mateos, *Super Resolution of Images and Video*. San Rafael, CA: Morgan & Claypool, 2007.
[17] R. Molina, M. Vega, J. Mateos, and A. K. Katsaggelos, "Variational posterior distribution approximation in Bayesian super resolution reconstruction of multispectral images," *Appl. Comput. Harmon. Anal.*, vol. 24, no. 2, pp. 251–267, Mar. 2008.
[18] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Total variation super resolution using a variational approach," in *Proc. Int. Conf. Image Process. (ICIP)*, San Diego, CA, Oct. 2008, pp. 641–644.
[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.
[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
[21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression (with discussion)," *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

[22] D. J. C. Mackay, "Probable networks and plausible predictions," *Network: Comput. Neural. Syst.*, vol. 6, no. 3, pp. 469–505, 1995.
[23] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," in *Advances in Neural Information Processing Syst. (NIPS) 20*. Cambridge, MA: MIT Press, 2008, pp. 1625–1632.
[24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, Nov. 1999.
[25] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference: Life after the EM algorithm," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.
[26] The USC-SIPI Image Database. [Online]. Available: http://sipi.usc.edu/database/
[27] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," Mitsubishi Electric Research Laboratories, 2001, Tech. Rep. TR2001-030.
[28] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
[29] T. M. Mitchell, "Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression." [Online]. Available: http://www.cs.cmu.edu/Etom/NewChapters.html

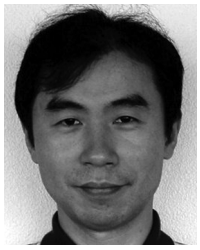


Atsunori Kanemura (M'09) received the M.Eng. degree in information science from Nara Institute of Science and Technology, Ikoma, Japan, in 2006, and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2009.

He is currently a Researcher at ATR Brain Information Communication Research Laboratory Group, Seika, Japan. He was an Intern at Google Inc., Mountain View, CA, a JSPS Research Fellow at Kyoto University, and a Research Fellow at the University of California, Santa Cruz, CA. His research interests include statistical image processing and machine learning.

Shin-ichi Maeda received the B.E. and M.E. degrees in electrical engineering from Osaka University, Osaka, Japan, in 1999 and 2001, respectively, and the Ph.D. degree in information science from Nara Institute of Science of Technology, Ikoma, Japan, in 2004.

He is currently an Assistant Professor at Kyoto University, Kyoto, Japan. His research interests are in machine learning, computational neuroscience, and coding theory.



Shin Ishii received the B.E. degree in chemical engineering, the M.E. degree in information engineering, and the Ph.D. degree in mathematical engineering, all from the University of Tokyo, Tokyo, Japan, in 1986, 1988, and 1997, respectively.

He is currently a Professor at Kyoto University, Kyoto, Japan. His research interests are computational neuroscience, systems neurobiology, and statistical learning theory.